Appendix

A NeurIPS Questionaire

1. Submission introducing new datasets must include the following in the supplementary materials:
   (a) Dataset documentation and intended uses. Recommended documentation frameworks include datasheets for datasets, dataset nutrition labels, data statements for NLP, and accountability frameworks.
   [Yes] We provide the complete 'datasheet for datasets' in Appendix B.
   (b) URL to website/platform where the dataset/benchmark can be viewed and downloaded by the reviewers.
   [Yes] `https://segmentmeifyoucan.com/`
   (c) Author statement that they bear all responsibility in case of violation of rights, etc., and confirmation of the data license.
   [Yes] All authors bear responsibility in case of violation of rights, etc. Confirmation of the data license is given the repository items of `https://zenodo.org/communities/segmentmeifyoucan`.
   (d) Hosting, licensing, and maintenance plan. The choice of hosting platform is yours, as long as you ensure access to the data (possibly through a curated interface) and will provide the necessary maintenance.
   [Yes] To ensure good availability, we chose professionally maintained platforms. Data is hosted at the public data repository `zenodo.com` and the benchmark website is hosted through `github.com`. Necessary maintenance such as updating the benchmark record etc. is shared between 3 different research groups such that there is always at least one person reachable.
2. To ensure accessibility, the supplementary materials for datasets must include the following:
   (a) Links to access the dataset and its metadata. This can be hidden upon submission if the dataset is not yet publicly available but must be added in the camera-ready version. In select cases, e.g when the data can only be released at a later date, this can be added afterward. Simulation environments should link to (open source) code repositories.
   [Yes] In general, all data is listed on `https://segmentmeifyoucan.com/`, and metadata more specifically in the zenodo mirrors: `https://zenodo.org/communities/segmentmeifyoucan`
   (b) The dataset itself should ideally use an open and widely used data format. Provide a detailed explanation on how the dataset can be read. For simulation environments, use existing frameworks or explain how they can be used.
   [Yes] The data is stored in standard formats: png, webp, json. We provide ready-to-use code that reads the data at `https://github.com/SegmentMeIfYouCan/road-anomaly-benchmark`.
   (c) Long-term preservation: It must be clear that the dataset will be available for a long time, either by uploading to a data repository or by explaining how the authors themselves will ensure this.
   [Yes] The data is uploaded to multiple mirrors, one of them is the public data repository `zenodo.org`.
   (d) Explicit license: Authors must choose a license, ideally a CC license for datasets, or an open source license for code (e.g. RL environments).
   [Yes] All images in the obstacle track were recorded by the authors of this work and are published under CC-BY 4.0 license. The images of the anomaly track are all publicly available and licensed as one of {public domain, CC-BY, CC-BY-SA}. A complete list of images, licenses and creators is published as part of the data record: `https://zenodo.org/record/5185336`.
   (e) Add structured metadata to a dataset's meta-data page using Web standards (like schema.org and DCAT): This allows it to be discovered and organized by anyone. If you use an existing data repository, this is often done automatically.
   [Yes] Metadata is part of the records on zenodo and accessible via different APIs, e.g. `https://zenodo.org/oai2d?verb=ListRecords&set=user-segmentmeifyoucan&metadataPrefix=oai_dc`.

(f) Highly recommended: a persistent dereferenceable identifier (e.g. a DOI minted by a data repository or a prefix on identifiers.org) for datasets, or a code repository (e.g. GitHub, GitLab,...) for code. If this is not possible or useful, please explain why.

- anomaly track data `https://doi.org/10.5281/zenodo.5185335`
- obstacle track data `https://doi.org/10.5281/zenodo.5186546`
- code repository is on GitHub `https://github.com/SegmentMeIfYouCan/road-anomaly-benchmark`

3. For benchmarks, the supplementary materials must ensure that all results are easily reproducible. Where possible, use a reproducibility framework such as the ML reproducibility checklist, or otherwise guarantee that all results can be easily reproduced, i.e. all necessary datasets, code, and evaluation procedures must be accessible and documented.
[Yes] While, as a public benchmark, we do not give access to the test labels and therefore nobody else should be able to produce the same measurements, we document all code that is used to create the benchmark results (directly yielding the *results.json* that is used for updating the public leaderboard on the website). Further, we created a small validation datasets that allows researchers to check that their method runs as intended. For these validation datasets, we report results in table 10 and table 11 which can be reproduced with the set of methods included in our benchmark suite.
4. For papers introducing best practices in creating or curating datasets and benchmarks, the above supplementary materials are not required. [N/A]

B Datasheet for Datasets

The following section is a complete answer to the datasheet questions from [48].

B.1 Motivation

- **For what purpose was the dataset created?** To evaluate and compare anomaly segmentation methods in driving scenes. Such evaluation enables conclusions on how good methods, usually tested on simpler datasets, are, but also facilitates specific method development for autonomous driving.
- **Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?** The authors of this work created this dataset to find answers to their research questions. In particular, there was no external party ordering or suggesting the creation of such a benchmark.
- **Who funded the creation of the dataset?** See section Acknowledgements.
- **Any other comments?** No.

B.2 Composition

- **What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?** The dataset comprise high resolution images of street scenes with unusual objects, which are all annotated on pixel-level. The objects appearing in the anomaly track data were **not** placed artificially and therefore represent naturally occurring anomalies in a global context. For the obstacle track, the objects were selected and placed by the authors, choosing from available objects that can reasonably appear on a street.
- **How many instances are there in total (of each type, if appropriate)?** 100 images for the anomaly track containing 262 ground truth components (+ 10 images for validation), 327 for the obstacle track containing 388 ground truth components (+ 85 images with hard weather or lightning conditions, + 30 images for validation).
- **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** The set of possibly occurring anomalies in driving scenes is boundless. The instances in this dataset are therefore a subset. For the anomaly track, they are a random sample of openly licensed, available images on the web. Therefore, they have a good geographic coverage. For the obstacle track, all images were taken in

Switzerland and Germany. They have a good coverage over weather and seasons, but are highly biased to European context for both the street background and the selected objects.

- **What data does each instance consist of?** Each data point is an RGB image and a corresponding segmentation map.
- **Is there a label or target associated with each instance?** Yes, our labelling policy is described in Section 3.1.
- **Is any information missing from individual instances?** No.
- **Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?** No.
- **Are there recommended data splits (e.g., training, development/validation, testing)?** Yes, our data is supposed to be used for testing only and should not be used for training. We supply a small validation split that enables local testing before submission to the benchmark.
- **Are there any errors, sources of noise, or redundancies in the dataset?** The annotations were created by humans and can therefore contain errors.
- **Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?** The dataset is self-contained.
- **Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctorpatient confidentiality, data that includes the content of individuals' non-public communications)?** No. All used images are licensed to be shared publicly.
- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** No.
- **Does the dataset relate to people?** People appear in some images of the Anomaly track.
- **Does the dataset identify any subpopulations (e.g., by age, gender)?** No.
- **Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?** It it possible to match faces in the dataset to any other database. However, this applies only to the images of the Anomaly track where all images used were already public, so our dataset did not change that. Regarding the images of Obstacle track identifying individuals is **not** possible.
- **Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?** No.
- **Any other comments?** No.

B.3 Collection Process

- **How was the data associated with each instance acquired?** In the obstacle track, the images were taken by the authors. For the anomaly track, openly licensed images from the web were collected. All images were annotated by humans.
- **What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?** Manual human curation.
- **If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?** Images in the anomaly track are a random sample of openly licensed, available images from the web, that show street scenes including at least one anomaly and are of sufficiently high quality. In the obstacle track, some images were extracted from sequences and only images at certain distances (at a rough guess) were included.
- **Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?**

The authors and student/research assistants. Everyone involved in the data generation process was employed at a university at the time of collecting and therefore drew a regular salary.

- **Over what timeframe was the data collected?** The images of the Anomaly Track were collected between August 2019 and August 2021. The images of the Obstacle track were collected between August 2020 and August 2021.
- **Were any ethical review processes conducted (e.g., by an institutional review board)?** No.
- **Does the dataset relate to people?** People appear in some images of the Anomaly track.
- **Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?** We obtained this data via third parties or other sources.
- **Were the individuals in question notified about the data collection?** The collected images were already licensed as public domain or creative commons, i.e., licensed to be shared and used.
- **Did the individuals in question consent to the collection and use of their data?** As we used images from the public domain or licensed a creative commons, we did not ask for consent ourselves.
- **Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis)been conducted?** Not beyond the Broader Impact section.
- **Any other comments?** No.

B.4 Preprocessing/cleaning/labeling

- **Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** The images for the anomaly track were resized and cropped to two different resolutions (1280×720 and 2048×1024).
- **Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** No.
- **Is the software used to preprocess/clean/label the instances available?** The open-source software ImageMagick was used for resizing the images. As labeling tool, LabelMe was used, which is publicly available (`https://github.com/wkentaro/labelme`).
- **Any other comments?** No.

B.5 Uses

- **Has the dataset been used for any tasks already?** Yes, for this paper.
- **Is there a repository that links to any or all papers or systems that use the dataset?** Yes, the public leaderboard on `https://segmentmeifyoucan.com/leaderboard`.
- **What (other) tasks could the dataset be used for?** No other task, since the labels are hidden.
- **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** No.
- **Are there tasks for which the dataset should not be used?** Certification of fittness for deployment would require at least a larger dataset.
- **Any other comments?** No.

B.6 Distribution

- **Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** Yes, the images including the labels for the validation set are public. The labels of the test set however will not be distributed.

- **How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?** Via multiple mirrors as zip archives, all listed on the website `https://segmentmeifyoucan.com/datasets`.
- **When will the dataset be distributed?** Now.
- **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** All parts that are distributed are under public domain or creative commons licenses.
- **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** No.
- **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** No.
- **Any other comments?** No.

B.7 Maintenance

- **Who is supporting/hosting/maintaining the dataset?** The authors of this paper.
- **How can the owner/curator/manager of the dataset be contacted (e.g., email address)?** `blumh@ethz.ch`,
- **Is there an erratum?** No.
- **Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?** In case that corrections are necessary, all versions are tracked in the `zenodo.com` data items.
- **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?** No. All images are licensed to be shared.
- **Will older versions of the dataset continue to be supported/hosted/maintained?** In case that there would be multiple versions, only the newest will be maintained.
- **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** Yes, we already have plans to incorporate another body of data into the benchmark. Similarly to the two existing datasets, each set of data is treated as a separate instance, but made comparable by using the same metrics. This can also be observed in this paper and the further comparisons that are listed in the Appendix.
- **Any other comments?** No.

C More Details on Evaluation Metrics

C.1 Pixel level

Let $\mathcal{Z}$ denote the set of image pixel locations. A model with a binary classifier providing scores $s(x) \in \mathbb{R}^{|\mathcal{Z}|}$ for an image $x \in \mathcal{X}$ (from a dataset $\mathcal{X} \subseteq [0,1]^{N \times |\mathcal{Z}| \times 3}$ of N images) discriminates between the two classes anomaly and non-anomaly. We evaluate the separability of the pixel-wise anomaly scores via the area under the precision-recall curve (AuPRC).

Let $\mathcal{Y} \subseteq \{\text{“anomaly”}, \text{“not anomaly”}\}^{N \times |\mathcal{Z}|}$ be the set of ground truth labels per pixel for $\mathcal{X}$. Analogously, we denote the predicted labels with $\hat{\mathcal{Y}}(\delta)$, obtained by pixel-wise thresholding on $s(x)\ \forall\ x \in \mathcal{X}$ w.r.t. some threshold value $\delta \in \mathbb{R}$. Then, for the anomaly class ($c_1 =$ “anomaly”) we compute

$$\text{precision}(\delta) = \frac{|\mathcal{Y}_{c_1} \cap \hat{\mathcal{Y}}_{c_1}(\delta)|}{|\hat{\mathcal{Y}}_{c_1}(\delta)|}, \quad \text{recall}(\delta) = \frac{|\mathcal{Y}_{c_1} \cap \hat{\mathcal{Y}}_{c_1}(\delta)|}{|\mathcal{Y}_{c_1}|} \tag{5}$$

with $\mathcal{Y}_{c_1}$ and $\hat{\mathcal{Y}}_{c_1}$ representing the ground truth labels and predicted labels, respectively. For the AuPRC, precision and recall are considered as functions of δ. The AuPRC approximates

$\int \text{precision}(\delta)\, d\text{recall}(\delta)$ and is threshold independent [49]. It also puts emphasis on detecting the minority class, making it particularly well suited as our main evaluation metric since the pixel-wise class distributions of RoadAnomaly21 and RoadObstacle21 are considerably unbalanced, *c.f.* section 3.1.

To consider the safety point of view, we also include the false positive rate at 95% true positive rate (FPR_{95}) in our evaluation, where the true positive rate (TPR) is equal to the recall of the anomaly class. The false positive rate (FPR) is the number of pixels falsely predicted as anomaly over the number of all non-anomaly pixels. Hence, for the anomaly class we compute

$$\text{FPR}_{95} = \frac{|\hat{\mathcal{Y}}_{c_1}(\delta') \cap \mathcal{Y}_{c_2}|}{|\mathcal{Y}_{c_2}|} \quad \text{s.t. } \text{TPR}(\delta') = 0.95\ , \tag{6}$$

where c_2 = "not anomaly". The metric FPR_{95} indicates how many false positive predictions are necessary to guarantee a desired true positive rate. Note that, any prediction which is contained in a ground truth labeled region of class void is not counted as false positive, *c.f.* section 3.1. In particular for the RoadObstacle21 dataset the evaluation is therefore restricted to the road area.

C.2 Component level - Qualitative examples revealing the difference of IoU and sIoU

If we consider component-level metrics over ground-truth components, it may happen that several components are close together and therefore covered by one predicted component. Although the real error can be small, the IoU punishes both ground-truth components. The same holds the other way around when considering metrics over predicted components, *i.e.* when one ground-truth component is covered by several predicted components. A qualitative example is given in figure 5. A small number of incorrectly predicted pixels may cause a strong decrease in the IoU. The adjusted IoU (sIoU) is less sensitive in such cases. sIoU focuses on correctly covering the regions of obstacles/anomalies in the image rather than finding such regions separately for each instance, as done by IoU. In self-driving it is more important to know the regions of anomaly rather than how many of them exist.

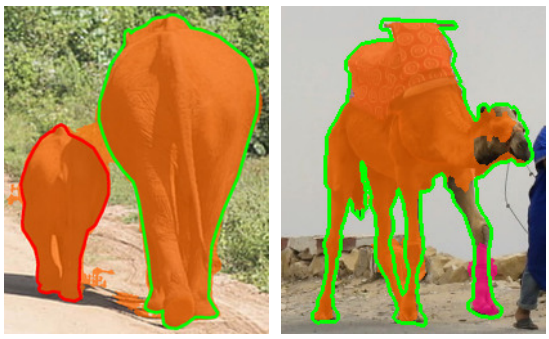


Figure 5: Two examples underlining the difference between IoU and adjusted IoU (sIoU). The ground-truth components are indicated by green/red contours, predicted components are highlighted by other colors. *Left*: Two ground-truth components (green & red) intersect with one predicted component (orange). Green: **IoU 68.18% vs. sIoU 87.01%**; red: **IoU 21.68% vs. sIoU 68.44%**. *Right*: Two predicted components (orange & pink) intersect with one ground-truth component (green). Orange: **IoU 78.97% vs. sIoU 81.69%**; pink: **IoU 03.44% vs. sIoU 18.91%**.

D Evaluated Methods

In this section, we first briefly introduce the methods which are evaluated on our benchmark and constitute our initial leader board. Afterwards we additionally provide technical details to those introduced methods.

D.1 Brief Description of Methods

All methods subject to evaluation are stated in boldface in the following. We evaluate at least one method per type discussed in section 2.2. All methods have an underlying semantic segmentation DNN trained on Cityscapes and they all provide pixel-wise anomaly scores.

Given an input image, the **maximum softmax probability** (MSP) of a DNN's corresponding output is a commonly-used baseline for OoD detection at image level [23]. Adding small perturbations to every pixel of the input image and applying temperature scaling enhances the anomaly detection ability of MSP. The latter approach is known as **ODIN** [22]. Another well-known method detects anomalies based on the **Mahalanobis distance**. It is computed by estimating Gaussian distributions of latent features of a DNN's penultimate layer, therefore yielding an estimate of the likelihood of a test sample w.r.t. the distribution in the training data. All these methods are originally designed for image classification but can be adapted straightforwardly to segmentation and represent good baselines in our benchmark.

As Bayesian approach to uncertainty estimation we employ **Monte Carlo (MC) dropout** in our evaluation. MC dropout has already been investigated for semantic segmentation. We follow [35] and use the mutual information as pixel-wise anomaly scores, which captures the epistemic uncertainty of a DNN. Furthermore, we additionally evaluate an **ensemble** of semantic segmentation networks.

In [7] several approaches to learning the confidence with respect to the presence of anomalies have been proposed. The **learned embedding density** aims to approximate the distribution of feature embeddings within a DNN via normalizing flows. At test time, the negative log-likelihood for each embedded representation of an image measures the discrepancy of a test embedding with respect to training embeddings, where high discrepancies indicate anomalies. These scores are then upsampled via bilinear interpolation to obtain the pixel-wise anomaly scores. Alternatively, the segmentation DNN can be modified to learn the confidence for the presence of anomalies, requiring an OoD dataset. As in [7], a Cityscapes DNN is trained with an additional model output for the Cityscapes void class. The anomaly scores are then the softmax scores for the that class, therefore this method is called **void classifier**. Additionally, one can also retrain a DNN with a different OoD proxy, such as the COCO dataset [4], and enforce **maximized softmax entropy** [11] on samples of the OoD proxy. All theses methods tune previously-trained DNNs to the task of anomaly segmentation and are included in our evaluation.

As autoencoders in our evaluation, we employ **image resynthesis** together with a discrepancy network that extracts meaningful differences based on the information provided by the DNN's segmentation mask, the resynthesized input image and the original image itself [14]. This approach can be extended by including uncertainty estimates in the discrepancy module, aiming to boost the anomaly segmentation performance, known as **SynBoost** [39]. One method specifically designed for obstacle segmentation is called **road inpainting** [42]. This method inpaints road patches in a sliding window manner. The resulting synthesized image is then again presented to a discrepancy network, similarly as in [14], for pixel-wise obstacle scores.

D.2 Method Description in Detail

All methods provide pixel-wise anomaly scores $s(x) \in \mathbb{R}^{|\mathcal{Z}|}, x \in \mathcal{X}$ where anomalies correspond to higher values. As a reminder, $\mathcal{Z}$ denotes the set of image coordinates and $\mathcal{X} \subseteq [0,1]^{N \times |\mathcal{Z}| \times 3}$ a dataset with N images. Below, we describe how s is obtained for each approach.

Maximum softmax probability. Let $f : \mathcal{X} \to \mathbb{R}^{|\mathcal{Z}| \times |\mathcal{C}|}$ denote the output of a semantic segmentation DNN. The maximum softmax probability (MSP) is a commonly-used baseline for OoD detection at image level [23]. It computes an anomaly score for each pixel $z \in \mathcal{Z}$ as

$$s_z(x) = 1 - \max_{c \in \mathcal{C}} \sigma(f_z^c(x)),\ x \in \mathcal{X}\,, \tag{7}$$

where $\sigma(\cdot) : \mathbb{R}^{|\mathcal{C}|} \to (0,1)^{|\mathcal{C}|}$ denotes the softmax function over the non-anomalous class set $\mathcal{C}$.

ODIN. Let $t \in \mathbb{R} \setminus \{0\}$ be a temperature scaling parameter and $\varepsilon \in \mathbb{R}$ a perturbation magnitude. Following [22] small perturbations are added to every pixel $z \in \mathcal{Z}$ of image x by

$$\tilde{x}_z = x_z - \varepsilon \text{sign}\left(-\frac{\partial}{\partial x_z} \log \max_{c \in \mathcal{C}} \sigma(f_z^c(x)/t)\right)\,. \tag{8}$$

Then, an anomaly score is obtained analogously to equation (7) via the MSP as

$$s_z(x) = 1 - \max_{c \in \mathcal{C}} \sigma(f_z^c(\tilde{x})/t)\,. \tag{9}$$

Mahalanobis distance. Let $h^{L-1}(\cdot)$ denote the output of the penultimate layer of a DNN with $L \in \mathbb{N}$ layers, *i.e.* $f(x) = h^L(x), x \in \mathcal{X}$. Under the assumption that

$$P(h_z^{L-1}(x) \mid y_z(x) = c) = \mathcal{N}(h_z^{L-1}(x) \mid \mu^c, \Sigma^c) \ , \tag{10}$$

an anomaly score for each pixel z can be computed as the Mahalanobis distance [21]

$$s_z(x) = \min_{c \in \mathcal{C}} \ (h_z^{L-1}(x) - \hat{\mu}^c)^T \hat{\Sigma}^{c^{-1}} (h_z^{L-1}(x) - \hat{\mu}^c) \ , \tag{11}$$

where $\hat{\mu}^c$ and $\hat{\Sigma}^c$ are estimates of the class mean μ^c and class covariance Σ^c, respectively, of the latent features in the penultimate layer. This Mahalanobis distance yields an estimate of the likelihood of a test sample with respect to the closest class distribution in the training data, which are assumed to be class-conditional Gaussians.

Monte Carlo dropout. Let $M \in \mathbb{N}$ denote the number of Monte Carlo sampling rounds and $\hat{q}_m^c := \sigma(f_z^c(x))$ the softmax probability of class $c \in \mathcal{C}$ for sample $m \in \{1, \dots, M\}$. The predictive entropy is computed as

$$\hat{E}(f(x)) = -\sum_{c \in \mathcal{C}} \left(\frac{1}{M} \sum_{m=1}^{M} \hat{q}_m^c \right) \log \left(\frac{1}{M} \sum_{m=1}^{M} \hat{q}_m^c \right) \ . \tag{12}$$

As suggested in [35], the mutual information can then be used to define an anomaly score

$$s_z(x) = \hat{E}(f(x)) - \frac{1}{M} \sum_{c \in \mathcal{C}} \sum_{m=1}^{M} \hat{q}_m^c \log \left(\hat{q}_m^c \right) \ . \tag{13}$$

Ensemble. Similar to Monte Carlo dropout, multiple samples of softmax probabilities $\hat{q}_m^c := \sigma(f_z^c(x)), c \in \mathcal{C}, m \in \{1, \dots, M\}$ are drawn from multiple semantic segmentation models. Those models have the same network architecture but are trained with different weights initialization [32]. Again, the mutual information is used as anomaly score

$$s_z(x) = \hat{E}(f(x)) - \frac{1}{M} \sum_{c \in \mathcal{C}} \sum_{m=1}^{M} \hat{q}_m^c \log \left(\hat{q}_m^c \right) \ . \tag{14}$$

Void classifier. In [36], an approach to learning the confidence with respect to the presence of anomalies was proposed. Here, we adapt this by using the Cityscapes void class to approximate the anomaly distribution. We then trained a Cityscapes DNN $f : \mathcal{X} \mapsto \mathbb{R}^{|\mathcal{Z}| \times (|\mathcal{C}|+1)}$ with an additional class, *i.e.*, a dustbin [5], and compute the anomaly score for each pixel $z \in \mathcal{Z}$ as the softmax score for the void class, which yields

$$s_z(x) = \sigma(f_z^{\text{void}}(x)), \ x \in \mathcal{X} \ . \tag{15}$$

Learned embedding density. Let $h^l(x) \in \mathbb{R}^{|\mathcal{Z}'| \times n_l}$, $n_l \in \mathbb{N}$, $\mathcal{Z}' \subset \mathcal{Z}$, be the embedding vector of a segmentation DNN at layer $l \in \{1, \dots, L\}$ for image $x \in \mathcal{X}$. The true distribution $p^*(h^l(x)), x \in \mathcal{X}_{\text{train}} \subset \mathcal{X}$ can be approximated with a normalizing flow $\hat{p}(h^l(x)) \approx p^*(h^l(x))$. At test time, the negative log-likelihood $-\log \hat{p}_{z'}(h^l(x)) \in (0, \infty)$ for each embedding location $z' \in \mathcal{Z}'$ then measures the discrepancy of a test embedding with respect to training embeddings, where higher discrepancies indicate anomalies [7]. The resulting anomaly score map are of size $|\mathcal{Z}'| = \frac{1}{n}|\mathcal{Z}|$, with $n \in \mathbb{N}$ the rescaling factor for $\mathcal{Z}'$ to match the size of $\mathcal{Z}$, and hence bring back latent features to the full image resolution $|\mathcal{Z}|$ via bilinear interpolation $u : \mathbb{R}^{|\mathcal{Z}'|} \rightarrow \mathbb{R}^{|\mathcal{Z}|}$. This yields an anomaly score for each $z \in \mathcal{Z}$ as

$$s_z(x) = u_z \left(\ (-\log \hat{p}(h_{z'}^l(x)))_{z' \in \mathcal{Z}'} \ \right), x \in \mathcal{X} \ . \tag{16}$$

Image resynthesis. The semantic segmentation map $\hat{y}(x) := (\arg\max_{c \in \mathcal{C}} f_z^c(x))_{z \in \mathcal{Z}}$ predicted by a DNN for image $x \in \mathcal{X}$ is passed to a generative network $g : \mathcal{C}^{|\mathcal{Z}|} \rightarrow \mathcal{X}'$ whose goal is to resynthesize x, *i.e.* $x \approx g(\hat{y}(x)) \in \mathcal{X}'$, with $\mathcal{X}'$ the resynthesized input space. Assuming that mislabeled pixels in the segmentation map, *i.e.* anomaly pixels, will be poorly reconstructed, a *discrepancy network* [14] $d : \mathcal{C}^{|\mathcal{Z}|} \times \mathcal{X}' \times \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{Z}|}$ is trained to extract the meaningful differences

based on the information provided by $\hat{y}(x), g(\hat{y}(x))$ and x itself. The output of $d(\cdot)$ serves as anomaly score for each $z \in \mathcal{Z}$, that is,

$$s_z(x) = d_z\left(\, \hat{y}(x), g(\hat{y}(x)), x \,\right),\ x \in \mathcal{X}\ . \tag{17}$$

Road inpainting. Another approach motivated by image resynthesis is road inpainting, which is specifically designed for obstacle segmentation. This method inpaints patches on the road (that is assumed to be known a-priori) in a sliding window manner and passes the resulting resynthesized image $g'(x)$ to the discrepancy network together with the original input image. Thus, the anomaly score is

$$s_z(x) = d_z\left(\, g'(x), x \,\right),\ x \in \mathcal{X}\ . \tag{18}$$

SynBoost. This approach follows a similar idea as image resynthesis but includes further inputs in the discrepancy module. In particular, for all $z \in \mathcal{Z}$ the pixel-wise softmax entropy

$$H_z(x) = -\sum_{c \in \mathcal{C}} \sigma(f_z^c(x)) \log\left(\sigma(f_z^c(x))\right) \tag{19}$$

and the pixel-wise softmax distance

$$D_z(x) = 1 - \max_{c \in \mathcal{C}} \sigma(f_z^c(x)) + \max_{c' \in \mathcal{C} \setminus \{\arg\max_{c \in \mathcal{C}}\}} \sigma(f_z^{c'}(x)) \tag{20}$$

are included. The anomaly score for $x \in \mathcal{X}$ is then obtained via

$$s_z(x) = d_z\left(\, \hat{y}(x), g(\hat{y}(x)), x, H(x), D(x) \,\right)\ . \tag{21}$$

Maximized entropy. Starting from a pretrained DNN, a second training objective is introduced to maximize the softmax entropy on OoD pixels [11, 13, 37]. This yields the multi-criteria loss function

$$(1-\lambda)\mathbb{E}_{(x,y)\sim\mathcal{D}_{in}}\left[\ell_{in}(\sigma(f_z(x)), y_z(x))\right] + \lambda\mathbb{E}_{x'\sim\mathcal{D}_{out}}\left[\ell_{out}(\sigma(f_z(x')))\right],\ \lambda \in [0,1]\ , \tag{22}$$

where ℓ_{in} is the empirical cross entropy and ℓ_{out} the averaged negative log-likelihood over all classes for the in-distribution data $\mathcal{D}_{in}$ and the out-distribution data $\mathcal{D}_{out}$, respectively. To approximate $\mathcal{D}_{out}$, a subset of the COCO dataset [4] is used whose images do not depict any object classes also available in $\mathcal{D}_{in}$, which is the Cityscapes dataset [1]. The COCO subset together with the Cityscapes training data are then included into a tender retraining of the pretrained Cityscapes model. The anomaly score is then computed via the softmax entropy as

$$s_z(x) = -\sum_{c \in \mathcal{C}} \sigma(f_z^c(x)) \log\left(\sigma(f_z^c(x))\right)\ . \tag{23}$$

D.3 Underlying Segmentation DNNs

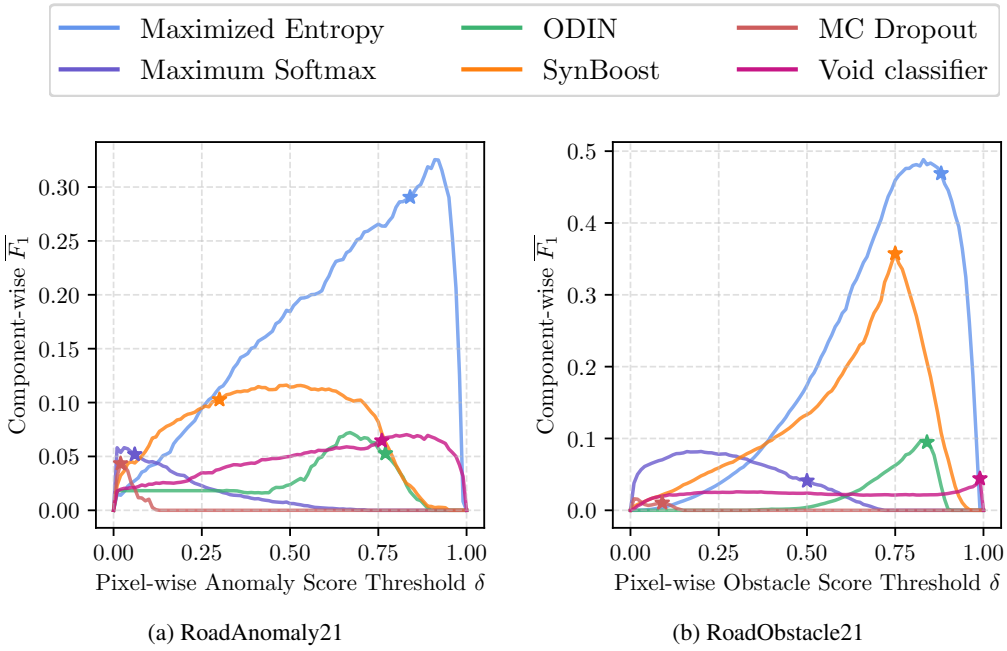
Most of our evaluated methods build upon variants of DeepLab [50] network architectures for semantic segmentation. In particular, for MC dropout, void classifier and learned embedding density we use a DeepLabv3+ model with an Xception backbone [51], as presented first in [7]. For maximum softmax, ODIN, Mahalanobis distance and maximized entropy, we employ a more modern DeepLabv3+ model with a WideResNet38 backbone [52]. For image resynthesis we use the more lightweight PSPNet as underlying model for semantic segmentation just like originally proposed by [14]. All these networks are initialized with publicly available weights which are pretrained on the Cityscapes dataset. To show the capacity of the network, we report the mean Intersection over Union (mIoU) on the Cityscapes validation dataset in Table 5.

D.4 Inference Time Comparison

In practice, anomaly segmentation is desired to be obtained in real time. Therefore, we report the run-time of the evaluated anomaly segmentation methods as further performance metric that expresses a method's suitability as online application. We measure the total inference time for RoadAnomaly21, *i.e.* the time from feeding all images through a model to obtaining pixel-wise anomaly scores. Afterwards we average the time per image and report them in table 5. All methods are compared with the same hardware (NVIDIA Quadro P6000), however they might differ in the underlying network architecture.

Method	Semantic segmentation Network architecture	mIoU ↑ on Cityscapes Val.	time in s ↓ per image
Maximum softmax	DeepLabv3+ WideResNet38 backbone [52]	90.3%	1.17
ODIN	DeepLabv3+ WideResNet38 backbone [52]	90.3%	16.74
Mahalanobis Distance	DeepLabv3+ WideResNet38 backbone [52]	90.3%	63.60
MC dropout	DeepLabv3+ Xcpection backbone [51]	80.3%	19.68
Void Classifier	DeepLabv3+ Xcpection backbone [51]	80.3%	2.02
Embedding density	DeepLabv3+ Xcpection backbone [51]	80.3%	10.66
Image resynthesis	PSPNet [53]	79.9%	1.43
SynBoost	DeepLabv3+ WideResNet38 backbone [52]	90.3%	2.09
Maximized entropy	DeepLabv3+ WideResNet38 backbone [52]	89.3%	1.07

Table 5: Run time comparison on a NVIDIA Quadro P6000 for different anomaly segmentation methods. The averaged inference time for one image of RoadAnomaly21 is reported in seconds. Moreover, the mean Intersection over Union (mIoU) on the Cityscapes validation dataset is reported to check whether anomaly segmentation decreases the original semantic segmentation performance.



(a) RoadAnomaly21

(b) RoadObstacle21

Figure 6: The averaged component-wise metric $\overline{F_1}$ as function of the pixel-wise anomaly / obstacle threshold δ for RoadAnomaly21 and RoadObstacle21, respectively, *c.f.* table 2 and 3. The "star" marker indicates a method's $\overline{F_1}$-score at the chosen threshold δ^* according to equation (4), which is used in our default procedure for generating segmentation masks from pixel-wise anomaly / obstacle scores. We observe that for most methods δ^* yields near optimal component-wise $\overline{F_1}$-scores, however not for every single tested method. Therefore, we encourage competitors to submit their own anomaly segmentation masks based on more sophisticated methods.

E Parameter Study

In our evaluation, the component-wise F_1 score (equation (3)) does not only depend on the parameter τ but also δ. Recall that τ is the threshold for sIoU at which one component is considered to be false negative and true positive, respectively, see also section 3.2. As we generate anomaly segmentation masks from pixel-wise anomaly scores, we introduced another threshold δ at which a given pixel is considered as anomaly. For generating segmentation masks with our default method, we chose that threshold as δ^* (equation (4)) which is the parameter for which a method achieves its best pixel-wise F_1 score, *i.e.* the optimal threshold according to the precision recall curve.

		Component-level metrics with filtering						Component-level metrics without filtering					
Method	OoD data	$k \in \mathcal{K}$ $\overline{\text{sIoU}} \uparrow$	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{PPV}} \uparrow$	$\tau = 0.50$ FN $\downarrow$	FP $\downarrow$	$F_1 \uparrow$	$\overline{F_1} \uparrow$	$k \in \mathcal{K}$ $\overline{\text{sIoU}} \uparrow$	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{PPV}} \uparrow$	$\tau = 0.50$ FN $\downarrow$	FP $\downarrow$	$F_1 \uparrow$	$\overline{F_1} \uparrow$
Maximum softmax [23]	✗	15.5	15.3	233	714	5.8	5.9	15.4	15.7	232	713	6.0	5.8
ODIN [22]	✗	19.6	17.9	226	985	5.6	6.0	19.7	17.5	227	983	5.5	6.0
Mahalanobis [21]	✗	14.8	10.2	241	1478	2.4	2.9	14.8	10.5	241	1464	2.4	2.9
MC dropout [35]	✗	20.5	17.3	225	1391	4.4	4.9	20.5	17.3	225	1391	4.4	4.9
Ensemble [32]	✗	16.4	20.8	233	1511	3.2	3.4	19.8	12.6	225	1528861	0.0	0.0
Void classifier [7]	✓	21.1	22.1	219	845	7.5	7.6	21.1	22.1	219	845	7.5	7.6
Embedding density [7]	✗	33.8	20.5	176	1485	9.4	9.2	34.0	20.8	176	1491	9.4	9.2
Image resynthesis [14]	✗	39.5	11.0	153	1225	13.7	12.9	39.6	11.1	152	1225	13.8	13.0
SynBoost [39]	✓	35.0	18.3	178	1114	11.5	11.5	34.7	17.8	179	1129	11.3	11.2
Maximized entropy [11]	✓	**49.2**	**39.5**	**115**	**421**	**35.4**	**34.5**	**49.2**	**39.4**	**115**	**421**	**35.4**	**34.4**

Table 6: Comparison of benchmark results for our RoadAnomaly21 dataset when not using the filtering included in our default segmentation post-processing step. This dataset contains 262 ground-truth components in total. The main performance metrics are highlighted with gray columns.

In this section, we perform a parameter study to show what impact the choice of δ has on the component-wise performance. By considering $\overline{F_1}$ as component-wise performance metric we already cover varying values for τ, since $\overline{F_1}$ is the average of component-wise F_1-scores over different values of τ. The dependence of $\overline{F_1}$ on the parameter δ is illustrated in figure 6 for RoadAnomaly21 and RoadObstacle21, respectively. For the sake of clarity, we only include six methods in total in this study, with at least one per type as discussed in section 2.2.

We observe that for most of the evaluated methods the choice of δ^* leads to an $\overline{F_1}$-score close its optimum, with some methods even reaching their optimal scores at δ^*, *e.g.* MC dropout on RoadAnomaly21 and SynBoost as well as the void classifier on RoadObstacle21. For the other methods the gap to the optimal $\overline{F_1}$-score reaches up to 2.8 percent points for maximized entropy on RoadAnomaly21 and even 4.1 percent points for maximum softmax on RoadObstacle21. However, except for the latter case where the distance between δ^* and the actual optimal location for $\overline{F_1}$ is 0.30, for all other methods the distance (in terms of $\overline{F_1}$) of δ^* to the optimal δ is at most 0.05.

This parameter study shows that our default method for generating segmentation masks from pixel-wise anomaly scores via the threshold δ^* is a legitimate choice, reaching a near optimal component-wise performance. Nonetheless, the parameter study also demonstrates that for some methods the $\overline{F_1}$-score can still be improved. Consequently, we allow (and encourage) competitors in the benchmark to submit their own anomaly segmentation masks with more sophisticated image operations and other post-processing techniques.

Another parameter included in the computation of the evaluation metrics is the size of predicted components in segmentation masks when generated from pixel-wise score maps. In our default post-processing method, we remove all components smaller than 500 pixels and 50 pixels in the anomaly and obstacle track, respectively, to reduce the amount of false positive components. To use this kind of filtering is completely optional. However, as can be seen in Table 6, Table 7, Table 8 and Table 9, we recommend using the post-processing option when competitors do not include a more sophisticated method. This is also why, we make our post-processing step transparent in this work since the size parameters are based on knowledge of ground truth components.

F Evaluated Datasets

Besides RoadAnomaly21 and RoadObstacle21 we also performed analogous benchmark evaluations for three additional publicly available datasets: Fishyscapes LostAndFound [7], LostAndFound test set [25], and the LiDAR guided Small obstacle Segmentation dataset [26]. For the sake of comparison, we chose the Fishyscapes LostAndFound validation set for the anomaly track and the LostAndFound test set as well as the Small Obstacle dataset for the obstacle track.

F.1 RoadAnomaly21 & RoadObstacle21 Validation Dataset

In order to ensure that methods run as intended with our benchmark code, we provide small validation sets (including ground truth annotations) for the anomaly track, called *RoadAnomaly21 validation*, and for the obstacle track, called *RoadObstacle21 validation*.

		Component-level metrics with filtering						Component-level metrics without filtering					
Method	OoD data	$k \in \mathcal{K}$ $\overline{\text{sIoU}} \uparrow$	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{PPV}} \uparrow$	$\tau = 0.50$ FN $\downarrow$	FP $\downarrow$	$F_1 \uparrow$	$\overline{F_1} \uparrow$	$k \in \mathcal{K}$ $\overline{\text{sIoU}} \uparrow$	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{PPV}} \uparrow$	$\tau = 0.50$ FN $\downarrow$	FP $\downarrow$	$F_1 \uparrow$	$\overline{F_1} \uparrow$
Maximum softmax [23]	✗	19.7	15.9	326	1503	6.3	6.9	21.5	8.1	325	9624	1.3	1.4
ODIN [22]	✗	20.7	18.5	312	1079	9.9	10.0	22.3	9.6	308	7260	2.1	2.1
Mahalanobis [21]	✗	14.0	21.8	352	1104	4.7	5.5	17.0	7.5	348	13630	0.6	0.7
MC dropout [35]	✗	6.3	5.8	375	2784	0.8	1.0	7.0	2.9	375	20727	0.1	0.1
Ensemble [32]	✗	8.6	4.7	365	3768	1.1	1.3	11.0	1.8	364	369439	0.0	0.0
Void classifier [7]	✓	6.3	20.3	365	350	6.0	5.9	2.8	**42.8**	384	**123**	1.6	2.6
Embedding density [7]	✗	35.6	2.9	244	11037	2.5	2.4	36.1	1.6	246	33598	0.8	0.8
Image resynthesis [14]	✗	16.6	20.5	334	773	8.9	9.5	17.4	15.8	332	7003	1.5	1.6
Road inpainting [42]	✗	**57.6**	39.5	131	586	41.8	40.2	**59.7**	17.2	**127**	4789	9.6	9.3
SynBoost [39]	✓	44.3	41.8	185	363	42.6	40.4	45.2	22.6	185	1432	20.1	19.2
Maximized entropy [11]	✓	47.9	**62.6**	**177**	**158**	**55.7**	**54.2**	48.7	35.1	177	758	**31.1**	**30.3**

Table 7: Comparison of benchmark results for our RoadObstacle21 dataset when not using the filtering included in our default segmentation post-processing step. This dataset contains 388 ground-truth components in total. The main performance metrics are highlighted with gray columns.

		Component-level metrics with filtering						Component-level metrics without filtering					
Method	OoD data	$k \in \mathcal{K}$ $\overline{\text{sIoU}} \uparrow$	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{PPV}} \uparrow$	$\tau = 0.50$ FN $\downarrow$	FP $\downarrow$	$F_1 \uparrow$	$\overline{F_1} \uparrow$	$k \in \mathcal{K}$ $\overline{\text{sIoU}} \uparrow$	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{PPV}} \uparrow$	$\tau = 0.50$ FN $\downarrow$	FP $\downarrow$	$F_1 \uparrow$	$\overline{F_1} \uparrow$
Maximum softmax [23]	✗	3.5	9.5	164	199	0.5	1.8	11.7	3.1	159	23134	0.1	0.1
ODIN [22]	✗	9.9	21.9	146	142	11.7	9.7	19.5	5.5	136	6113	0.9	0.9
Mahalanobis [21]	✗	19.6	29.4	132	147	19.1	19.2	28.9	8.8	124	4009	1.9	2.1
MC dropout [35]	✗	4.8	18.1	160	120	3.4	4.3	8.7	14.8	158	1835	0.7	0.9
Ensemble [32]	✗	3.1	1.1	162	1643	0.3	0.4	6.6	0.5	156	226622	0.0	0.0
Void classifier [7]	✓	9.2	39.1	149	**38**	14.6	14.9	9.6	16.6	149	**304**	6.6	6.6
Embedding density [7]	✗	5.9	10.8	155	202	5.3	4.9	12.1	5.7	150	3990	0.7	0.7
Image resynthesis [14]	✗	5.1	12.6	157	191	4.4	4.1	6.3	6.0	157	5875	0.3	0.3
SynBoost [39]	✓	**27.9**	**48.6**	**107**	62	**40.7**	**38.0**	**35.3**	**16.6**	**97**	723	**14.2**	**13.3**
Maximized entropy [11]	✓	21.1	**48.6**	121	56	33.2	30.0	27.1	12.1	113	1160	7.6	6.9

Table 8: Comparison of benchmark results for the Fishyscapes LostAndFound validation dataset when not using the filtering included in our default segmentation post-processing step. This dataset contains 165 ground-truth components in total. The main performance metrics are highlighted with gray columns.

These datasets show similar scenes and objects as in RoadAnomaly21 test and RoadObstacle21 test, respectively. The splits contain 10 images with 16 ground truth components and 30 images with 45 ground truth objects in total, respectively. Note that although both datasets share the same setup as in the corresponding test splits, they are still **not** representative for the test data since they contains only a very limited number of different road surfaces and diverse obstacle types. Therefore we do not recommend to fine-tune methods on these two validation datasets.

Moreover, we applied our set of anomaly segmentation methods to RoadObstacle21 validation, see table 11. Some of those methods are also made publicly available in our benchmark code to compare to and reproduce the reported results.

F.2 Fishyscapes LostAndFound

The Fishyscapes LostAndFound validation dataset [7] consists of 100 images from the original LostAndFound data [25] with refined labels. With this labeling, anomalous objects are not restricted to only appear on the road but everywhere in the image, therefore Fishyscapes LostAndFound fits our benchmark's anomaly track.

Comparing the RoadAnomaly21 and Fishyscapes LostAndFound datasets in terms of anomaly class frequency per pixel location, as observed in figure 10, one notices a clear difference in the variation of object locations and sizes. While in Fishyscapes LostAndFound the objects appear mostly in the center of the image and are also rather small, the objects in RoadAnomaly21 may appear everywhere in the image and have sizes ranging from 122 up to 883,319 pixels (thus covering up to more than one third of the image). The low variety in object sizes is also noticeable in the pixel-wise class distribution, as in RoadAnomaly21 13.8% of the pixels belong to the anomaly class and 82.2% to non-anomaly whereas in Fishyscapes LostAndFound only 0.23% belong to anomaly and 81.13% to non-anomaly.

		Component-level metrics with filtering						Component-level metrics without filtering					
Method	OoD data	$k \in \mathcal{K}$ sIoU ↑	$\hat{k} \in \hat{\mathcal{K}}$ PPV ↑	$\tau = 0.50$ FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑	$k \in \mathcal{K}$ sIoU ↑	$\hat{k} \in \hat{\mathcal{K}}$ PPV ↑	$\tau = 0.50$ FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [23]	✗	14.2	62.2	1575	602	11.0	13.4	16.3	17.5	1572	31481	0.8	1.1
ODIN [22]	✗	38.9	48.0	971	1303	39.4	38.1	40.2	29.9	967	5962	17.6	17.2
Mahalanobis [21]	✗	33.8	31.7	1126	2314	25.3	24.6	34.7	22.8	1124	7677	11.7	11.6
MC dropout [35]	✗	17.0	34.7	1453	1641	14.2	14.7	17.7	20.0	1451	9560	4.5	4.7
Ensemble [32]	✗	6.7	7.6	1604	5649	2.8	2.7	7.5	3.8	1600	299431	0.1	0.1
Void classifier [7]	✓	0.7	35.1	1698	**108**	1.2	1.1	0.7	25.1	1698	**351**	1.1	1.0
Embedding density [7]	✗	37.8	35.2	963	1973	33.7	30.8	38.6	18.9	961	6862	16.1	14.8
Image resynthesis [14]	✗	27.2	30.7	1232	2093	22.3	21.5	28.0	19.7	1228	15418	5.5	5.3
Road inpainting [42]	✗	**49.2**	60.7	**749**	646	**57.9**	**56.9**	**50.4**	33.0	**743**	4852	25.7	25.2
SynBoost [39]	✓	37.2	**72.3**	930	230	57.3	53.0	37.6	**63.3**	931	535	**51.5**	**47.7**
Maximized entropy [11]	✓	45.9	63.1	781	598	57.4	55.0	46.7	35.8	778	2813	34.1	32.7

Table 9: Comparison of benchmark results for the LostAndFound test-NoKnown dataset when not using the filtering included in our default segmentation post-processing step. This dataset contains 1709 ground-truth components in total. The main performance metrics are highlighted with gray columns.

As already discussed in section 4, we observe a less pronounced gap between methods designed for image classification and those specifically designed for anomaly segmentation. A detailed overview of our benchmark results on Fishyscapes LostAndFound is given in table 12. In this evaluation, we see that the number of false positive components (relative to the number of ground truth components) over multiple thresholds τ is significantly less than on RoadAnomaly21, shown in Table 2. This holds for all evaluated methods, resulting in relatively strong component-wise performance (compared to SynBoost and maximized entropy). Even Mahalanobis and void classifier report strong results, which is due to similarity of this dataet to Cityscapes as all LostAndFound images share the same setup as in Cityscapes. These results further indicate the lack in diversity in Fishyscapes LostAndFound. More specifically, the environments of the scenes shown in LostAndFound do not considerably differ to those shown in Cityscapes whereas our RoadAnomaly21 dataset has a wide variety of scenes since all images are gathered from the web, see figure 13.

F.3 LostAndFound test-NoKnown

The LostAndFound dataset [25] shares the same setup as Cityscapes but includes small obstacles on the road. Therefore, this dataset fits our benchmark's obstacle track. When a model is trained on Cityscapes, the LostAndFound dataset then contains images with objects that have been previously seen and therefore are not anomalies. As most of our methods are designed for anomaly detection, we filtered out all scenes in the LostAndFound test split where the obstacles belong to known classes, *e.g*. children or bicycles, and call this subset LostAndFound test-NoKnown. In this way, the results obtained with our evaluated methods on LostAndFound test-NoKnown and on our RoadObstacle21 dataset are comparable.

Both datasets have obstacles in the same size range. Both RoadObstacle21 and LostAndFound test-NoKnown have 0.12% of the pixels labeled as obstacles, while 39.08% and 15.31% of the pixels belong to not obstacles, respectively. Regarding the object locations in images, the obstacles in RoadObstacle21 are distributed wider over the image than in LostAndFound, as observed in figure 10. This also implies that in RoadObstacle21 the obstacles appear at stronger varying distances. For an illustration as well as of that variation, we refer to figures 15 and 16. Looking at the results in table 13, we observe for LostAndFound test-NoKnown, just as in Fishyscapes LostAndFound (table 12), that methods from image classification perform relatively well in comparison to methods designed for anomaly segmentation. This is again due the limited variety of environments, *i.e*. the road surfaces in this dataset. In our RoadObstacle21 dataset, we therefore provide scenes with obstacles on different road surfaces, such as gravel or a road with cracks, see figure 16.

Regarding the dataset size, LostAndFound achieves their high number of images by densely sampling from video sequences. Consequently, some images depict nearly identical scenes (same environment and obstacle combination with the obstacle approximately at the same distance), see *e.g*. Figure 17. In RoadObstacle21 the number of different environment and obstacle combinations is considerably higher due to the wide variety of 31 object types in the dataset. If multiple images depict the same scene, we made sure that the distance to the obstacle (and therefore the size of the obstacle in the image) varies noticeably from image to image, *c.f*. Figure 18.

F.4 LiDAR Guided Small Obstacle Dataset

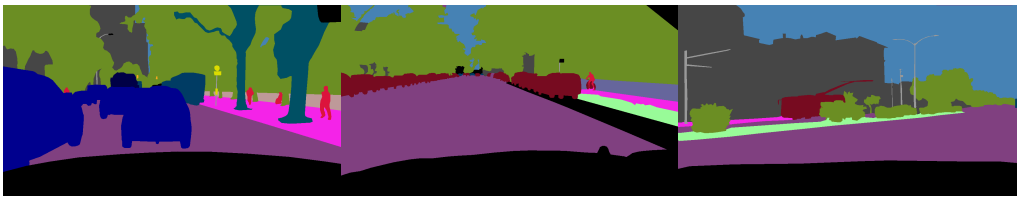
The third publicly available dataset to which we applied our benchmark suite is the LiDAR guided Small obstacle Segmentation dataset [26], which can be viewed as a reference dataset for our obstacle track. The results corresponding to this dataset are given in table 14. In general, the given set of methods exhibits poor performance on this dataset. More precisely, obstacles are mostly overlooked, *e.g*. SynBoost as best-performing method still misses 1100 of 1203 components in total at the lowest sIoU threshold $\tau = 0.25$. As the LiDAR guided Small obstacle Segmentation dataset rather focuses on the challenge of detecting obstacles via multiple sensors, including LiDAR, the camera images of this dataset are purposely challenging, *e.g*. due to low illumination, blurry images and barely visible obstacles. Figure 7 shows an example of this dataset, which highlights the difficulty of anomaly detection. This dataset can easily be included into our benchmark and it also fits the obstacle track, however, from our experiments we conclude that this dataset is less suitable to camera-only obstacle segmentation as obstacles are not well captured via cameras.

(a) Input image (b) Ground truth (c) Maximized entropy

Figure 7: An example image (a) from the Small Obstacle dataset with the corresponding ground truth annotation (b) and an obstacle score heatmap obtained with maximized entropy (c). Here, the obstacles are barely visible in the input image due to their size and the scene's illumination, that is why camera-only based segmentation techniques tend to fail for this dataset.

F.5 CAOS BDD-Anomaly

The CAOS BDD-Anomaly dataset [8] consists of images sourced from BDD100k [2]. In order to create an anomaly segmentation dataset, the authors split the BDD100k data such that images with motorcycles, bicycles and trains are separated from the rest. These left out objects are then considered as anomalies. We do not perform any experiments on CAOS BDD-Anomaly since the considered anomalous objects are not strictly unknown. They also appear in Cityscapes [1] on which most semantic segmentation models are trained. Moreover, we find several labeling mistakes that hinder proper evaluation of anomaly segmentation performance, see figure 8.



(a) Vegetation as train (b) Cars as bicycle (c) Bus as bicycle

Figure 8: Some examples of labeling mistakes in the CAOS BDD-Anomaly dataset, where in-distribution objects are incorrectly annotated as anomaly, *i.e*. train and bicycle.

G Evaluation per Environment Category

We already emphasized that in our RoadObstacle21 dataset a wide variety of road surfaces are available, representing different scenes which might pose unique challenges. In this section, we provide more insights by evaluating our set of methods on each of these surfaces. In total, we split our datasets into 9 different scenes, shown in figure 9:

1. cracked road, surrounded by snow (road cracked)
2. dark asphalt after rain, with leaves (asphalt dark)
3. gravel road, no snow (road gravel)
4. gray asphalt in village and forest (asphalt gray)
5. motorway with side railing (motorway)
6. sun reflection off wet road (sun reflection)
7. road made of bricks (road bricks)
8. night images (asphalt night)
9. and snowstorm images .

Figure 9: The scenes of our RoadObstacle21 dataset feature a variety of road surfaces.

We evaluate each subset using our benchmark suite and report the results in table 16. This more detailed evaluation shows that the reported set of methods perform differently across the data splits, with no method having consistent performance on each of these subsets. Our dataset offers extra difficulty caused by the diversity of road texture, surrounding environments, weather and lighting variations. Cracks and leaves may trigger false positives, and a gravel or wet road surface may itself be sufficiently different from training images to be mistaken for an anomaly.

H Evaluation for Different Component Sizes

In this section we provide further insights of the segmentation quality of ground truth components in RoadAnomaly21 and RoadObstacle21. To this end, we conduct a more fine-grained analysis by grouping ground truth components into size intervals and perform the evaluation for each size interval separately. In total, RoadAnomaly21 contains 259 ground truth components, ranging in size from 122 to 883,319 pixels. RoadObstacle21 contains 388 obstacles ranging from 18 up to 77,435 pixels. For each dataset we divide these components into eight size intervals such that each interval contains same number of components.

In figure 11, we report the averaged sIoU (equation (1)) w.r.t. the ground truth components within each size interval. As illustrated in this figure, we observe a positive correlation of sIoU with the component size. Especially in RoadObstacle21, methods designed for the task of anomaly segmentation like maximized entropy or SynBoost perform significantly better than the other approaches.

In addition, we consider the amount of entirely neglected components, meaning the objects for which not even one pixel is detected. To do so, we measure the relative ratio of FN to all ground truth components within different object size intervals, see figure 12. As a threshold, therefore, for discriminating between FN and TP, we choose $\tau = 0$, *i.e.* a ground truth component is considered as TP if at least one of its pixels is detected by the respective method. Indeed, we observe a negative correlation of the number of FN with the component size, but even more conspicuous is the amount of totally overlooked components of small size. This analysis shows the challengingness of anomaly segmentation, particularly for small obstacles at component-level, and emphasizes the need for further research in this direction.

I Evaluation per Object Category

As part of our benchmark, we also provide an evaluation with respect to different object categories. An exemplary evaluation with the given set of methods is provided in table 15. In particular, the methods specifically designed for anomaly segmentation perform worse on the vehicle category than on the other ones. This general trends shows that our choice of vehicles, including classes such as jet ski, rickshaw and carriage, is rather challenging. This additional dimension of granularity offers further insight to users of our benchmark such that one can identify the drawbacks of an anomaly segmentation method under inspection.

Tables and Figures

		Pixel-level			Component-level											
	requires	Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	$\tau = 0.25$			$\tau = 0.50$			$\tau = 0.75$			
Method	OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{PPV}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [23]	✗	40.4	60.2	42.6	34.2	18.6	7	37	29.0	10	39	19.7	15	44	3.3	16.7
ODIN [22]	✗	46.3	61.5	49.2	37.1	24.5	6	37	31.7	10	38	20.0	16	47	0.0	16.4
Mahalanobis [21]	✗	22.5	86.4	31.7	17.8	11.7	12	80	8.0	16	84	0.0	16	86	0.0	2.4
MC dropout [35]	✗	29.2	77.9	35.3	26.6	16.7	9	82	13.3	12	84	7.7	16	88	0.0	5.7
Ensemble [32]	✗	16.0	80.0	30.3	20.9	23.1	11	59	12.5	15	63	2.5	15	64	2.5	4.7
Void classifier [7]	✓	39.3	66.1	42.7	25.2	27.4	9	33	25.0	13	34	11.3	16	38	0.0	11.7
Embedding density [7]	✗	51.9	60.0	54.1	48.1	24.4	3	102	19.8	7	105	13.8	15	117	1.5	12.4
Image resynthesis [14]	✗	76.4	20.5	72.0	46.8	25.3	2	66	29.2	9	68	15.4	15	81	2.0	15.0
SynBoost [39]	✓	68.8	30.9	65.6	46.7	21.9	3	65	27.7	9	70	15.1	15	76	2.2	15.7
Maximized entropy [11]	✓	**80.7**	**17.4**	**74.3**	**63.6**	**45.0**	**1**	**24**	**54.5**	**3**	**25**	**48.1**	**11**	**29**	**20.0**	**41.6**

Table 10: Benchmark results for the RoadAnomaly21 validation set. This dataset contains 16 ground truth components.

		Pixel-level			Component-level											
	requires	Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	$\tau = 0.25$			$\tau = 0.50$			$\tau = 0.75$			
Method	OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{PPV}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [23]	✗	43.4	3.8	53.7	40.6	20.9	20	113	27.3	22	113	25.4	32	117	14.9	22.5
ODIN [22]	✗	46.6	4.0	48.4	38.7	28.4	23	84	29.1	23	84	29.1	33	88	16.6	26.2
Mahalanobis [21]	✗	25.9	26.1	27.7	28.3	27.3	24	111	23.7	33	114	14.0	41	120	4.7	14.8
MC dropout [35]	✗	7.9	43.8	13.4	8.9	8.7	39	245	4.1	42	246	2.0	45	246	0.0	2.0
Ensemble [32]	✗	4.7	98.3	9.2	4.1	57.9	42	8	10.7	44	8	3.7	45	8	0.0	4.6
Void classifier [7]	✓	9.8	43.6	15.6	11.2	24.2	39	35	14.0	39	35	14.0	42	36	7.1	11.9
Embedding density [7]	✗	1.5	56.7	3.4	15.9	2.8	31	1261	2.1	39	1262	0.9	45	1269	0.0	1.0
Image resynthesis [14]	✗	70.3	1.3	61.3	28.8	22.4	25	105	23.5	31	108	16.8	43	117	2.4	15.1
Road inpainting [42]	✗	90.4	98.9	**89.0**	52.9	**67.0**	17	**14**	64.4	18	**14**	62.8	**22**	**16**	**54.8**	61.6
SynBoost [39]	✓	81.4	2.8	73.2	37.0	43.6	18	25	55.7	28	30	37.0	39	37	13.6	38.4
Maximized entropy [11]	✓	**94.4**	**0.4**	88.4	**56.4**	60.8	**12**	19	**68.0**	**13**	19	**66.7**	24	21	48.3	**62.3**

Table 11: Benchmark results for the RoadObstacle21 validation set. This dataset contains 45 ground truth objects.

		Pixel-level			Component-level											
	requires	Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	$\tau = 0.25$			$\tau = 0.50$			$\tau = 0.75$			
Method	OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{PPV}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [23]	✗	5.6	40.5	12.1	3.5	9.5	152	199	6.9	164	199	0.5	165	199	0.0	1.8
ODIN [22]	✗	15.5	38.4	22.5	9.9	21.9	139	142	15.6	146	142	11.7	163	143	1.3	9.7
Mahalanobis [21]	✗	32.9	**8.7**	37.3	19.6	29.4	111	145	29.7	132	147	19.1	155	157	6.0	19.2
MC dropout [35]	✗	14.4	47.8	20.0	4.8	18.1	149	120	10.6	160	120	3.4	164	121	0.7	4.3
Ensemble [32]	✗	0.3	90.4	0.7	3.1	1.1	159	1643	0.7	162	1643	0.3	163	1643	0.2	0.4
Void classifier [7]	✓	11.7	15.3	21.9	9.2	39.1	143	**38**	19.6	149	**38**	14.6	158	**38**	6.7	14.9
Embedding density [7]	✗	8.9	42.2	14.8	5.9	10.8	148	202	8.9	155	202	5.3	163	202	1.1	4.9
Image resynthesis [14]	✗	5.1	29.8	11.1	5.1	12.6	150	190	8.1	157	191	4.4	164	191	0.6	4.1
SynBoost [39]	✓	**64.9**	30.9	**67.6**	**27.9**	**48.6**	**103**	62	**42.9**	**107**	62	**40.7**	**130**	63	**26.6**	**38.0**
Maximized entropy [11]	✓	44.3	37.7	50.9	21.1	**48.6**	117	56	35.7	121	56	33.2	146	57	15.8	30.0

Table 12: Benchmark results for the Fishyscapes LostAndFound validation set. This dataset contains 165 ground truth objects.

		Pixel-level			Component-level												
			Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	$\tau = 0.25$			$\tau = 0.50$			$\tau = 0.75$			
Method	requires OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{PPV}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [23]	✗	30.1	33.2	32.5	14.2	62.2	1256	580	33.0	1575	602	11.0	1701	644	0.7	13.4
ODIN [22]	✗	51.0	30.7	54.3	38.9	48.0	713	1271	50.1	971	1303	39.4	1349	1372	20.9	38.1
Mahalanobis [21]	✗	55.0	12.9	54.8	33.8	31.7	777	2146	38.9	1126	2314	25.3	1527	2585	8.1	24.6
MC dropout [35]	✗	36.2	36.0	42.2	17.0	34.7	1214	1626	25.8	1453	1641	14.2	1635	1674	4.3	14.7
Ensemble [32]	✗	2.9	82.0	8.2	6.7	7.6	1523	5633	4.9	1604	5649	2.8	1695	5705	0.4	2.7
Void classifier [7]	✓	4.4	47.0	13.7	0.7	35.1	1689	**108**	2.2	1698	**108**	1.2	1708	**109**	0.1	1.1
Embedding density [7]	✗	61.7	10.4	61.7	37.8	35.2	646	1873	45.8	963	1973	33.7	1526	2299	8.7	30.8
Image resynthesis [14]	✗	57.1	8.8	55.1	27.2	30.7	947	1990	34.2	1232	2093	22.3	1560	2304	7.2	21.5
Road inpainting [42]	✗	**83.0**	35.7	**79.1**	**49.2**	60.7	**631**	635	63.0	**749**	646	**57.9**	**958**	727	**47.1**	**56.9**
SynBoost [39]	✓	81.8	**4.6**	75.2	37.2	**72.3**	767	203	**66.0**	930	230	57.3	1378	436	26.7	53.0
Maximized entropy [11]	✓	77.9	9.7	76.8	45.9	63.1	639	589	63.5	781	598	57.4	1113	681	39.9	55.0

Table 13: Benchmark results for the LostAndFound test-NoKnown dataset. This dataset contains 1709 ground truth objects.

		Pixel-level			Component-level												
			Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	$\tau = 0.25$			$\tau = 0.50$			$\tau = 0.75$			
Method	requires OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{PPV}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [23]	✗	0.7	57.1	2.2	0.5	1.5	1196	1652	0.5	1202	1653	0.1	1203	1653	0.0	0.2
ODIN [22]	✗	1.7	51.7	5.7	2.7	3.9	1151	1829	3.4	1176	1834	1.8	1197	1841	0.4	1.9
Mahalanobis [21]	✗	1.4	**45.5**	2.4	7.1	4.0	1039	4863	5.3	1137	4882	2.1	1198	4907	0.2	2.4
MC dropout [35]	✗	0.5	82.2	2.1	0.5	2.8	1191	1406	0.9	1200	1407	0.2	1203	1408	0.0	0.3
Void classifier [7]	✓	0.8	59.6	2.1	1.5	4.9	1169	813	3.3	1193	816	1.0	1200	819	0.3	1.5
Embedding density [7]	✗	0.5	66.0	1.1	9.8	1.8	1010	12421	2.8	1122	12502	1.2	1200	12587	0.0	1.3
SynBoost [39]	✓	**12.5**	62.8	**22.8**	**11.5**	**14.4**	**1009**	1204	**14.9**	**1040**	1217	**12.6**	**1116**	1234	**6.9**	**12.0**
Maximized entropy [11]	✓	4.9	63.1	11.6	2.0	9.7	1159	**586**	4.8	1184	**586**	2.1	1202	**586**	0.1	2.4

Table 14: Benchmark results for the LiDAR guided Small obstacle Segmentation dataset. This dataset contains 1203 ground truth components in total.

		all anomalies			animals			vehicles			other anomalies		
		$N = 100$			$N = 59$			$N = 23$			$N = 11$		
Method	OoD data	AuPRC ↑	FPR_{95} ↓	$\overline{F_1}$ ↑	AuPRC ↑	FPR_{95} ↓	$\overline{F_1}$ ↑	AuPRC ↑	FPR_{95} ↓	$\overline{F_1}$ ↑	AuPRC ↑	FPR_{95} ↓	$\overline{F_1}$ ↑
Maximum softmax [23]	✗	28.0	72.0	5.4	25.2	75.6	4.9	30.2	71.9	5.0	26.1	57.7	9.3
ODIN [22]	✗	31.1	71.7	5.2	32.1	72.9	4.9	30.6	74.0	4.9	35.5	61.7	9.7
Mahalanobis [21]	✗	20.0	87.0	2.7	21.3	87.4	2.5	16.7	87.5	1.8	34.9	66.1	12.5
MC dropout [35]	✗	28.9	69.5	4.3	24.8	74.0	3.2	35.2	72.2	4.5	20.1	62.3	15.2
Ensemble [32]	✗	17.7	91.1	3.4	16.7	91.3	2.9	18.8	89.7	1.1	10.4	85.6	4.8
Void classifier [7]	✓	36.6	63.5	6.5	32.2	66.9	4.0	42.3	39.2	8.5	23.1	70.3	21.7
Embedding density [7]	✗	37.5	70.8	7.9	43.9	63.2	8.4	30.3	88.4	3.4	24.2	58.4	21.0
Image resynthesis [14]	✗	52.3	25.9	12.5	51.4	26.5	16.4	57.8	25.6	6.1	40.4	55.1	12.9
SynBoost [39]	✓	56.4	61.9	10.0	54.7	66.2	10.3	57.8	61.7	7.4	43.1	62.6	21.4
Maximized entropy [11]	✓	**85.5**	**15.0**	**28.7**	**92.2**	**7.2**	**41.9**	**79.0**	**17.8**	**16.2**	**51.6**	**18.3**	**25.4**

Table 15: Effect of different anomalies in the RoadAnomaly21 dataset. In total, RoadAnomaly21 contains 59 images with only animals, 23 images with only vehicles and 11 with other anomalies (*e.g.* cones, tents, ...). The number of images according to a subset is denoted with N in the table. Images containing objects from both the animal and the vehicle category (7 images in total) are excluded in this evaluation.

Method	OoD data	road cracked $N = 40$ AuPRC	road cracked $\overline{F_1}$	asphalt dark $N = 47$ AuPRC	asphalt dark $\overline{F_1}$	road gravel $N = 33$ AuPRC	road gravel $\overline{F_1}$	asphalt gray $N = 66$ AuPRC	asphalt gray $\overline{F_1}$	motorway $N = 30$ AuPRC	motorway $\overline{F_1}$	sun reflection $N = 72$ AuPRC	sun reflection $\overline{F_1}$	road bricks $N = 39$ AuPRC	road bricks $\overline{F_1}$
Maximum softmax [23]	✗	11.7	3.2	69.3	25.7	39.5	21.0	43.4	14.9	4.8	0.8	2.1	4.4	32.7	26.8
ODIN [22]	✗	14.9	4.8	74.8	30.8	65.3	37.0	73.8	22.5	9.9	7.2	2.8	5.4	48.8	22.0
Mahalanobis [21]	✗	25.9	1.6	46.7	18.3	65.8	21.9	84.7	53.8	61.2	35.6	13.9	0.5	83.6	41.0
MC dropout [35]	✗	6.5	1.0	21.7	4.9	27.9	5.5	11.4	1.7	0.6	0.0	0.2	0.2	18.5	3.6
Ensemble [32]	✗	34.3	0.0	5.6	0.8	33.4	0.0	17.3	12.3	1.2	4.3	0.2	0.0	17.6	0.6
Void classifier [7]	✓	15.9	6.4	35.0	15.4	6.3	3.1	38.2	11.0	18.7	8.4	10.7	0.6	13.4	10.1
Embedding density [7]	✗	2.5	0.8	3.3	0.8	2.7	2.2	1.8	2.4	1.1	3.0	0.1	1.1	18.3	2.7
Image resynthesis [14]	✗	48.2	9.6	42.0	12.3	77.0	42.4	66.6	22.2	23.7	12.9	34.4	9.0	12.1	2.4
Road inpainting [42]	✗	21.0	18.4	77.0	47.2	88.4	74.7	93.5	**79.8**	**83.1**	**78.1**	29.4	22.0	93.5	**73.7**
SynBoost [39]	✓	46.1	14.7	89.3	66.5	84.7	54.5	81.2	54.0	53.8	48.8	43.1	25.4	89.8	70.3
Maximized entropy [11]	✓	**77.1**	**42.5**	**96.9**	**71.9**	**98.6**	**88.7**	**94.8**	70.2	64.3	35.1	**43.2**	**30.6**	**93.9**	61.0

Method	OoD data	asphalt night $N = 30$ AuPRC	asphalt night $\overline{F_1}$	snowstorm $N = 55$ AuPRC	snowstorm $\overline{F_1}$
Maximum softmax [23]	✗	6.0	2.5	1.6	0.8
ODIN [22]	✗	8.0	1.8	6.7	4.6
Mahalanobis [21]	✗	14.2	5.5	21.2	13.2
MC dropout [35]	✗	4.2	1.1	0.5	0.6
Ensemble [32]	✗	11.5	16.9	0.6	0.0
Void classifier [7]	✓	5.9	5.5	3.0	5.1
Embedding density [7]	✗	16.7	3.6	0.9	2.6
Image resynthesis [14]	✗	16.5	6.3	19.2	4.0
Road inpainting [42]	✗	**51.2**	**28.0**	**55.3**	**35.0**
SynBoost [39]	✓	14.5	10.2	46.4	20.7
Maximized entropy [11]	✓	41.0	12.1	30.5	17.5

Table 16: Effect of different of scenes in the RoadObstacle21 dataset. Here, N denotes the number of images in a subset. As main evaluation metrics we consider the pixel-wise AuPRC and the component-wise $\overline{F_1}$.

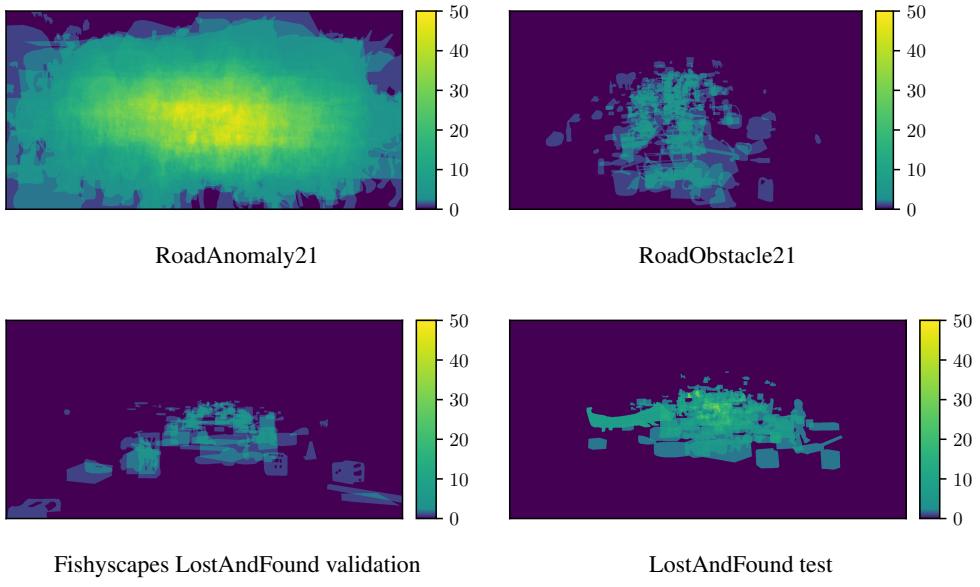
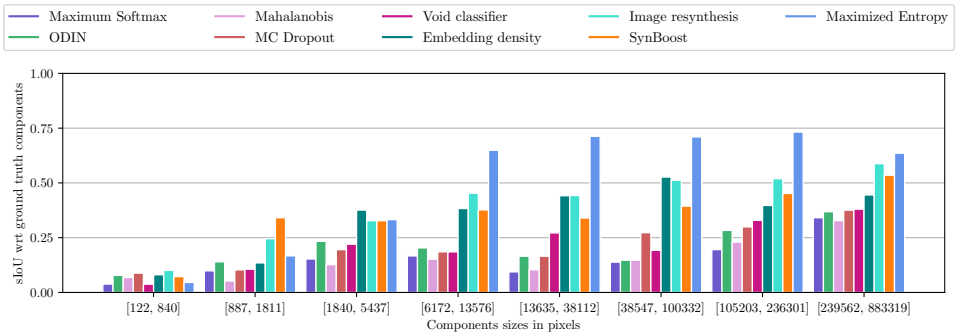
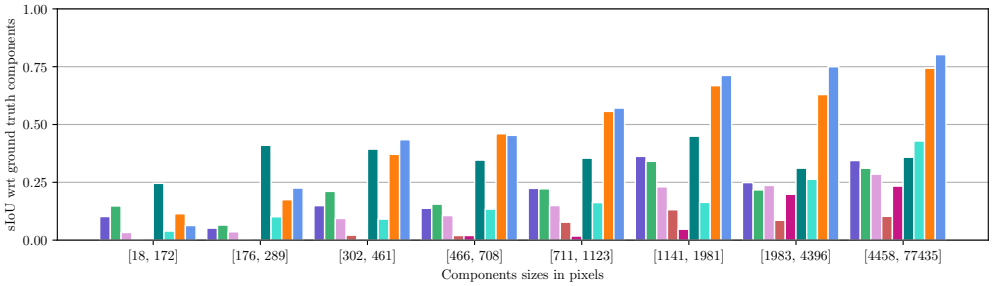


Figure 10: Comparison of the (spatial) pixel distributions between RoadAnomaly21 and Fishyscapes LostAndFound (100 images each) as well as RoadObstacle21 and a subset of randomly sampled images from the LostAndFound test dataset (327 images each). The color indicates the frequency of observing an anomaly in each pixel location, averaged over the images in the dataset.

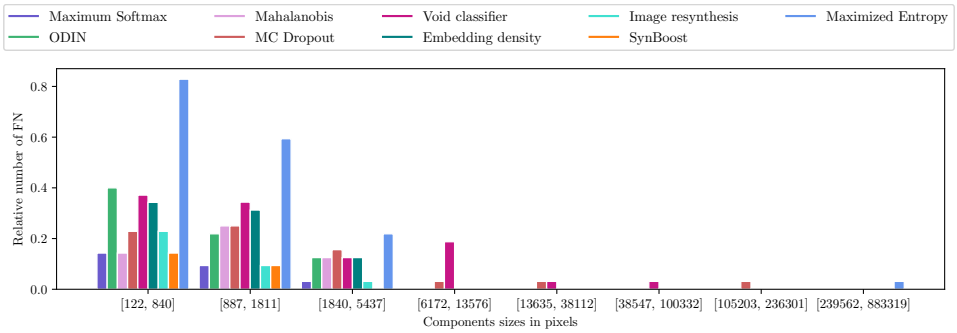


RoadAnomaly21, each size interval contains 32 components except the first one (very left) that contains 35 components

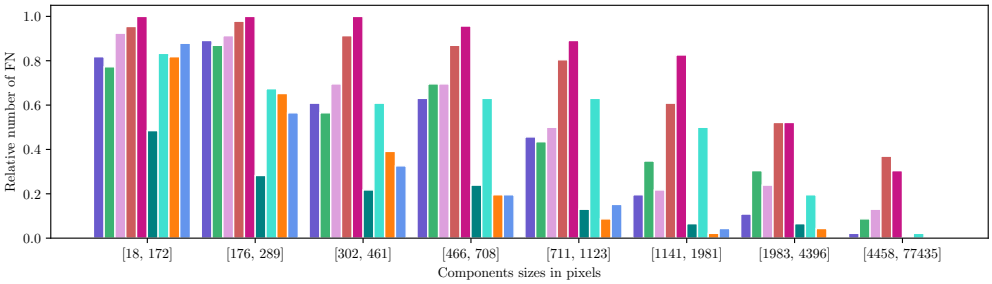


RoadObstacle21, each size interval contains 46 components except the first one (very left) that contains 66 components

Figure 11: Comparison of the averaged sIoU w.r.t. ground truth components within a certain range of the components size, produced by the methods discussed in section 3.3 and appendix D.2.



RoadAnomaly21, each size interval contains 32 components except the first one (very left) that contains 35 components



RoadObstacle21, each size interval contains 46 components except the first one (very left) that contains 66 components

Figure 12: Comparison of the relative number of FN to TP at threshold $\tau = 0$, *i.e.* the fraction of overlooked components to the total number of ground truth components within a certain range of the components size. The evaluated methods are discussed in section 3.3 and appendix D.2.

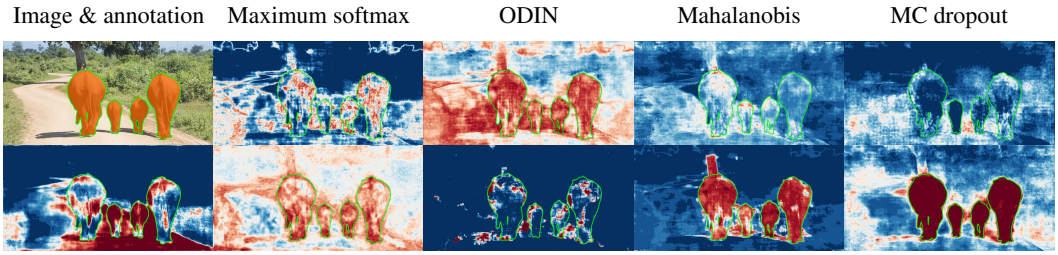


Figure 13: Qualitative comparison of the methods introduced in section 3.3 and appendix D.2 on a sample from RoadAnomaly21. In this example, the anomalous objects have a large size and the environment differs from scenes shown in Cityscapes. Green contours indicate the annotation of the anomaly.

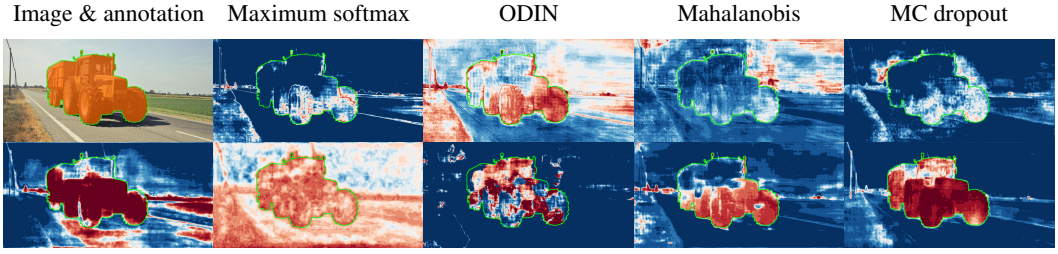


Figure 14: Qualitative comparison of the methods introduced in section 3.3 and appendix D.2 on a sample from RoadAnomaly21. The scene shows a tractor which does not appear in Cityscapes. Green contours indicate the annotation of the anomaly.

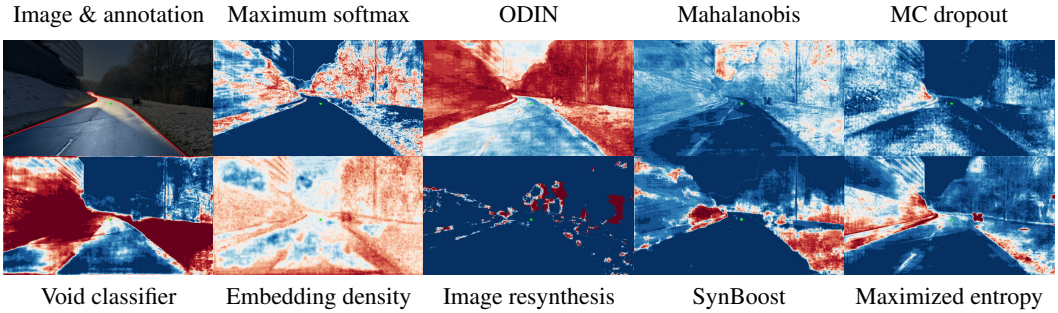


Figure 15: Qualitative comparison of the methods introduced in section 3.3 and appendix D.2 for an example from RoadObstacle21, where the obstacle is small and far away. Green contours indicate the annotation of the obstacle, red contours the road.

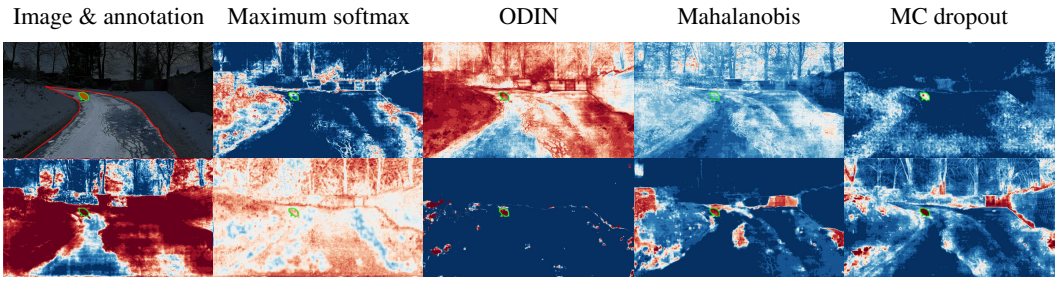


Figure 16: Qualitative comparison of the methods introduced in section 3.3 and appendix D.2 for an example from RoadObstacle21, showing a road surface with cracks. Green contours indicate the annotation of the obstacle, red contours the road.

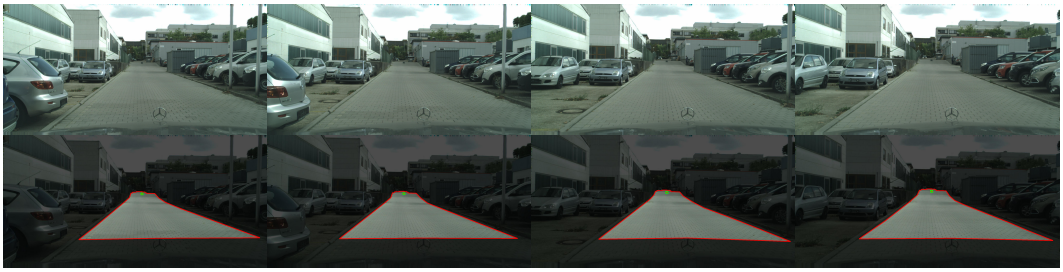


Figure 17: Four example images of densely sampled frames from a video sequence (of 18 frames in total) with ground truth annotation in the LostAndFound test set. Due to this sampling, LostAndFound achieve their high number of images but, as shown in this figure, several images are nearly identical.

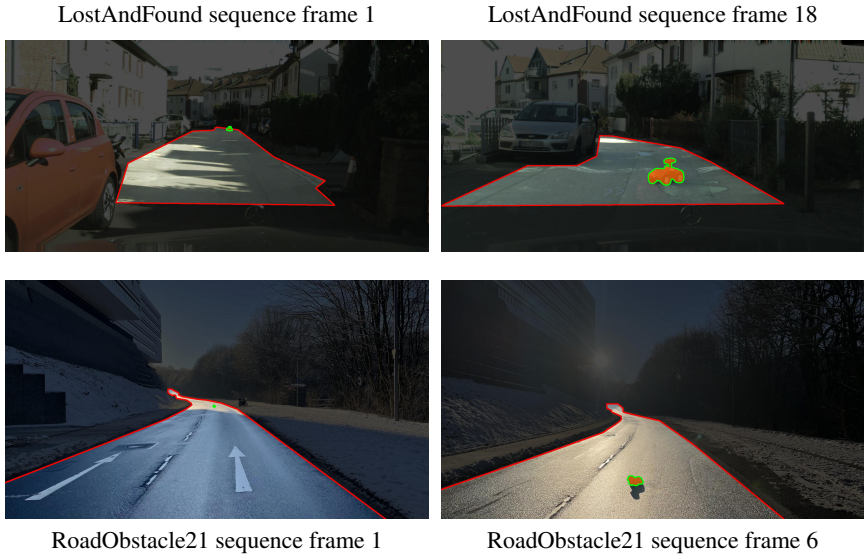


Figure 18: Comparison of one sequence from the LostAndFound test set (top row) and one sequence from the RoadObstacle21 test set (bottom row). In this figure, the first and last frame of a video sequence which are included in the respective test set, are shown. We observe that in this LostAndFound example 18 images of one sequence are included in the test while in RoadObstacle21 at most 6 frames are included (which differ significantly in lighting in this example).